

NETWORK ADDRESS TRANSLATION (NAT)



Karya Ilmiah
Disampaikan Pada Seminar Kenaikan Jabatan
Di Jurusan Teknik Komputer Politeknik Negeri Sriwijaya
Pada Tanggal 26 Agustus 2006

Oleh :

ALI FIRDAUS, S.Kom.
NIP. 132 297 884

POLITEKNIK NEGERI SRIWIJAYA
2006

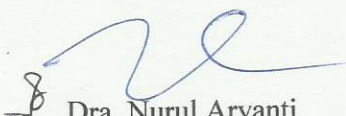
NETWORK ADDRESS TRANSLATION (NAT)

Oleh :

Ali Firdaus, S.Kom.
NIP. 132 297 884

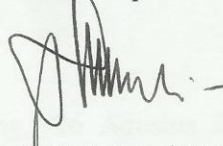
Palembang, 26 Agustus 2006

Menyetujui
Kepala Pusat Penelitian Dan
Pengabdian Kepada Masyarakat



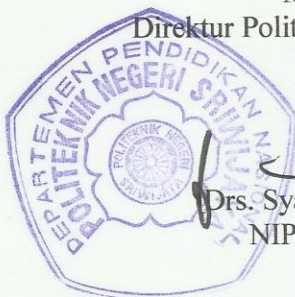
Dra. Nurul Aryanti
NIP. 132 064 906

Ketua Jurusan
Teknik Komputer



Ir. A. Bahri Joni Malyan
NIP. 131 196 014

Mengetahui
Direktur Politeknik Negeri Sriwijaya



Drs. Syamsul Bahri, MM
NIP. 131 803 780

DAFTAR ISI

	halaman
HALAMAN JUDUL	i
HALAMAN PENGESAHAN	ii
ABSTRAK	iii
KATA PENGANTAR	iv
DAFTAR ISI	v
DAFTAR GAMBAR	vi
BAB I PENDAHULUAN	
1.1. Latar Belakang	1
1.2. Perumusan Masalah	2
1.3. Tujuan dan Manfaat	2
BAB II TINJAUAN PUSTAKA	
2.1. TCP / IP	3
2.2. Gateway / Router.....	4
2.3. Fire Wall	4
2.4. DIVERT (mekanisme diversifikasi paket <i>kernel</i>).....	5
2.5. <i>Network Address Translation</i> (NAT).....	5
BAB III PEMBAHASAN	
3.1. Mekanisme NAT	6
3.2. Perbedaan Sistem NAT Dengan Sistem Proxy	7
3.3. Instalasi Natd	8
3.4. Konfigurasi TCP / IP Client	11
BAB IV PENUTUP	12
DAFTAR PUSTAKA	13

BAB I PENDAHULUAN

1.1. Latar Belakang

Pada mulanya Internet adalah sebagai jaringan komunikasi non-profit dan Internet didesain tanpa memperhatikan dunia bisnis. Perkembangan teknologi informasi pada saat ini semakin cepat seiring perkembangan teknologi lainnya baik dibidang hardware maupun software. Hal ini juga telah menyebabkan terjadinya sedikit perubahan pada kebudayaan masyarakat kita dengan ditandai adanya kesadaran masyarakat tentang pentingnya informasi yang cepat, tepat, dan selalu terbaru.

Internet sebagai salah satu sarana dan cara untuk mendapatkan informasi yang paling murah, cepat, dan selalu baru. Setiap orang bisa memilih dan mendapatkan banyak informasi yang akan dicari dari berbagai sumber. Tempat yang menyediakan jasa pelayanan internetpun sudah cukup banyak, dan bukan hanya di kota-kota besar saja, bahkan penyedia jasa internetpun pada saat ini mulai masuk didaerah-daerah kota kecamatan.

Kemudian hal ini menjadi masalah dimasa sekarang dan dimasa depan. Dengan semakin banyaknya penghuni Internet, baik pencari informasi maupun penyedia informasi, maka kebutuhan akan pengalamatan di Internet semakin membesar.

Permasalahan diatas cukup membuat dilema bagi penyedia jasa layanan internet, terutama dikalangan pendidikan yang memiliki dana terbatas, dan kalangan komersial yang akan mengurangi keuntungan yang seharusnya didapat. Keadaan demikian tentu sangat tidak kondusif, karena lalu arus lintas informasi kepada masyarakat akan sedikit terhambat dan pendapatan usaha penyedia jasa layanan internet pun berkurang.

Kebutuhan besar akan IP *address* biasanya terjadi di jaringan komputer perusahaan dan LAN di lembaga pendidikan. IP *address* sebagai sarana pengalamatan di Internet semakin menjadi barang mewah dan eksklusif. Tidak sembarang orang ini bisa mendapatkan IP *address* yang valid dengan mudah.

Berdasarkan latar belakang di atas maka penulis mencobamengkaji tentang *Network Address Translation (NAT): Cara untuk menghemat IP Address*

1.2. Perumusan Masalah

Permasalahan yang muncul disini adalah bagaimana cara melakukan penghematan IP *address* dengan salah satu mekanisme *Network Address Translation (NAT)*.

1.3. Tujuan Dan Manfaat

Tujuan penulisan ini adalah untuk mengetahui langkah-langkah sederhana yang dapat diterapkan untuk melakukan penghematan IP *address* namun tetap menjaga agar nomor IP *address* valid.

Melalui penulisan ini diharapkan bisa didapat cara untuk melakukan penghematan IP *address*, dan beberapa komputer bisa mengakses Internet walau hanya memiliki satu IP *address* yang valid.

BAB II TINJAUAN PUSTAKA

Sebelum kita membahas lebih lanjut ada baiknya kita urai kembali konsep-konsep dasar yang harus dipahami sebelum masuk ke *Network Address Translation* (NAT), Diantaranya adalah TCP/IP, *Gateway/Router*, dan *Firewall*.

2.1. TCP/IP

Protokol yang menjadi standar dan dipakai hampir oleh seluruh komunitas Internet adalah TCP/IP (*Transmission Control Protocol/Internet Protocol*). Agar komputer bisa berkomunikasi dengan komputer lainnya, maka menurut aturan TCP/IP, komputer tersebut harus memiliki suatu *address* yang unik. Alamat tersebut dinamakan *IP address*, “Douba (1995:6)”. *IP Address* memiliki format sbb: *aaa.bbb.ccc.ddd*. Contohnya: 167.205.19.33

Yang penting adalah bahwa untuk berkomunikasi di Internet, komputer harus memiliki *IP address* yang legal. Legal dalam hal ini artinya adalah bahwa alamat tersebut dikenali oleh semua *router* di dunia dan diketahui bahwa alamat tersebut tidak ada duplikatnya di tempat lain. *IP address* legal biasanya diperoleh dengan menghubungi InterNIC.

Suatu jaringan internal bisa saja menggunakan *IP address* sembarang. Namun untuk tersambung ke Internet, jaringan itu tetap harus menggunakan *IP address* legal. Jika masalah *routing* tidak dibereskan (tidak menggunakan *IP address* legal), maka saat sistem kita mengirim paket data ke sistem lain, sistem tujuan itu tidak akan bisa mengembalikan paket data tersebut, sehingga komunikasi tidak akan terjadi.

Dalam berkomunikasi di Internet/antar jaringan komputer dibutuhkan *gateway/router* sebagai jembatan yang menghubungkan simpul-simpul antar jaringan sehingga paket data bisa diantar sampai ke tujuan, “Douba (1995:116)”

2.2. Gateway/Router

Gateway adalah komputer yang memiliki minimal 2 buah *network interface* untuk menghubungkan 2 buah jaringan atau lebih, “*Unix Integration to WAN (2000:35)*”. Di Internet suatu alamat bisa ditempuh lewat *gateway-gateway* yang memberikan jalan/rute ke arah mana yang harus dilalui supaya paket data sampai ke tujuan. Kebanyakan *gateway* menjalankan *routing daemon* (program yang meng-*update* secara dinamis tabel *routing*). Karena itu *gateway* juga biasanya berfungsi sebagai *router*. *Gateway/router* bisa berbentuk *Router box* seperti yang di produksi Cisco, 3COM, dll, atau bisa juga berupa komputer yang menjalankan *Network Operating System* plus *routing daemon*, “*FreeBSD Handbook, (1998:97)*”. Misalkan PC yang dipasang **Unix FreeBSD** dan menjalankan program Routed atau Gated. Namun dalam pemakaian Natd, *routing daemon* tidak perlu dijalankan, jadi cukup dipasang *gateway* saja.

Karena *gateway/router* mengatur lalu lintas paket data antar jaringan, maka di dalamnya bisa dipasangi mekanisme pembatasan atau pengamanan (*filtering*) paket-paket data. Mekanisme ini disebut *Firewall*.

2.3. Firewall

Sebenarnya *Firewall* adalah suatu program yang dijalankan di *gateway/router* yang bertugas memeriksa setiap paket data yang lewat kemudian membandingkannya dengan *rule* yang diterapkan dan akhirnya memutuskan apakah paket data tersebut boleh diteruskan atau ditolak. Tujuan dasarnya adalah sebagai *security* yang melindungi jaringan internal dari ancaman dari luar. Namun dalam tulisan ini *Firewall* digunakan sebagai basis untuk menjalankan *Network Address Translation (NAT)*.

Dalam FreeBSD, program yang dijalankan sebagai *Firewall* adalah ipfw. Sebelum dapat menjalankan ipfw, *kernel GENERIC* harus dimodifikasi supaya mendukung fungsi *firewall*. Ipfw mengatur lalu lintas paket data berdasarkan IP asal, IP tujuan, nomor *port*, dan jenis *protocol*. Untuk menjalankan NAT, option IPDIVERT harus diaktifkan dalam *kernel*.

2.4. **DIVERT (mekanisme diversi paket kernel)**

Socket divert sebenarnya sama saja dengan *socket* IP biasa, kecuali bahwa *socket divert* bisa di *bind* ke *port divert* khusus lewat *bind system call*. IP address dalam *bind* tidak diperhatikan, hanya nomor *port*-nya yang diperhatikan. Sebuah *socket divert* yang di *bind* ke *port divert* akan menerima semua paket yang di *diversikan* pada *port* tersebut oleh mekanisme di *kernel* yang dijalankan oleh implementasi *filtering* dan program *ipfw*. Mekanisme ini yang dimanfaatkan nantinya oleh *Network Address Translator*, "Tito@TI.ITB.ac.id (2005 :2)".

2.5. **Network Address Translation (NAT)**

Dalam FreeBSD, mekanisme *Network Address Translation* (NAT) dijalankan oleh program *Natd* yang bekerja sebagai *daemon*. *Network Address Translation Daemon* (*Natd*) menyediakan solusi untuk permasalahan penghematan ini dengan cara menyembunyikan IP address jaringan internal, dengan membuat paket yang di-generate di dalam terlihat seolah-olah dihasilkan dari mesin yang memiliki IP address legal, "FreeBSD Handbook, (1998:201). *Natd* memberikan konektivitas ke dunia luar tanpa harus menggunakan IP address legal dalam jaringan internal.

Natd menyediakan fasilitas *Network Address Translation* untuk digunakan dengan *socket divert*. *Natd* mengubah semua paket yang ditujukan ke *host* lain sedemikian sehingga *source* IP addressnya berasal dari mesin *Natd*. Untuk setiap paket yang diubah berdasarkan aturan ini, dibuat tabel translasi untuk mencatat transaksi ini.

Dengan NAT, aturan bahwa untuk berkomunikasi harus menggunakan IP address legal, dilanggar. NAT bekerja dengan jalan mengkonversikan IP-IP address ke satu atau lebih IP address lain. IP address yang dikonversi adalah IP address yang diberikan untuk tiap mesin dalam jaringan internal (bisa sembarang IP). IP address yang menjadi hasil konversi terletak di luar jaringan internal tersebut dan merupakan IP address legal yang *valid/routable*.

BAB III PEMBAHASAN

3.1. Mekanisme NAT

Sebuah paket TCP terdiri dari *header* dan data. *Header* memiliki sejumlah *field* di dalamnya, salah satu *field* yang penting di sini adalah MAC (*Media Access Control*) *address* asal dan tujuan, IP *address* asal dan tujuan, dan nomor *port* asal dan tujuan.

Saat mesin A menghubungi mesin B, *header* paket berisi IP A sebagai IP *address* asal dan IP B sebagai IP *address* tujuan. *Header* ini juga berisi nomor *port* asal (biasanya dipilih oleh mesin pengirim dari sekumpulan nomor *port*) dan nomor *port* tujuan yang spesifik, misalnya *port* 80 (untuk *web*).

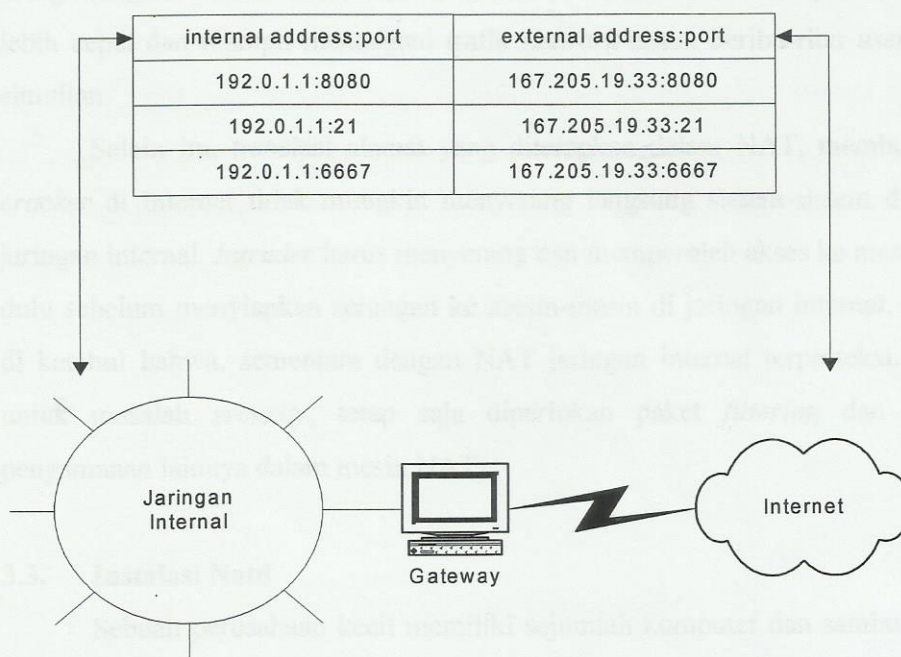
Kemudian B menerima paket pada *port* 80 dan memilih nomor *port* balasan untuk digunakan sebagai nomor *port* asal menggantikan *port* 80 tadi. Mesin B lalu membalik IP *address* asal & tujuan dan nomor *port* asal & tujuan dalam *header* paket. Sehingga keadaan sekarang IP B adalah IP *address* asal dan IP A adalah IP *address* tujuan. Kemudian B mengirim paket itu kembali ke A. Selama *session* terbuka, paket data hilir mudik menggunakan nomor *port* yang dipilih.

Router (yang biasa – tanpa *Natd*) memodifikasi *field* MAC *address* asal & tujuan dalam *header* ketika me-*route* paket yang melewatinya. IP *address*, nomor *port*, dan nomor *sequence* asal & tujuan tidak disentuh sama sekali. NAT juga bekerja atas dasar ini. Dimulai dengan membuat tabel translasi internal untuk semua IP *address* jaringan internal yang mengirim paket melewatinya. Lalu men-*set* tabel nomor *port* yang akan digunakan oleh IP *address* yang valid. Ketika paket dari jaringan internal dikirim ke *Natd* untuk disampaikan keluar, *Natd* melakukan hal-hal sebagai berikut:

1. Mencatat IP *address* dan *port* asal dalam tabel translasi
2. Menggantikan nomor IP asal paket dengan nomor IP dirinya yang valid

3. Menetapkan nomor *port* khusus untuk paket yang dikirim keluar, memasukkannya dalam tabel translasi dan menggantikan nomor *port* asal tersebut dengan nomor *port* khusus ini.

Ketika paket balasan datang kembali, Natd mengecek nomor *port* tujuannya. Jika ini cocok dengan nomor *port* yang khusus telah ditetapkan sebelumnya, maka dia akan melihat tabel translasi dan mencari mesin mana di jaringan internal yang sesuai. Setelah ditemukan, ia akan menulis kembali nomor *port* dan IP address tujuan dengan IP address dan nomor *port* asal yang asli yang digunakan dulu untuk memulai koneksi. Lalu mengirim paket ini ke mesin di jaringan internal yang dituju. Natd memelihara isi tabel translasi selama koneksi masih terbuka.



Gambar 3.1 Mekanisme Natd

3.2. Perbedaan Sistem NAT Sistem Proxy

Hampir mirip dengan NAT, suatu jaringan kecil dengan *proxy* bisa menempatkan beberapa mesin untuk mengakses *web* dibelakang sebuah mesin yang memiliki IP address valid. Ini juga merupakan langkah penghematan biaya

dibanding harus menyewa beberapa account dari ISP dan memasang modem & sambungan telepon pada tiap mesin.

Namun demikian, *proxy* server ini tidak sesuai untuk jaringan yang lebih besar. Bagaimanapun, menambah *hard disk* dan RAM pada server *proxy* supaya *proxy* berjalan efisien tidak selalu dapat dilakukan (karena *constraint* biaya). Lagi pula, persentase *web page* yang bisa dilayani oleh *cache proxy* akan makin menurun sejalan dengan semakin menipisnya ruang kosong di *hard disk*, sehingga penggunaan *cache proxy* menjadi tidak lebih baik dari pada sambungan langsung. Tambahan lagi, tiap koneksi bersamaan akan meng-*generate* proses tambahan dalam *proxy*. Tiap proses ini harus menggunakan *disk I/O channel* yang sama, dan saat *disk I/O channel* jenuh, maka terjadilah *bottle neck*.

NAT menawarkan solusi yang lebih fleksibel dan *scalable*. NAT menghilangkan keharusan mengkonfigurasi *proxy/sock* dalam tiap *client*. NAT lebih cepat dan mampu menangani trafik *network* untuk beribu-ribu *user* secara simultan.

Selain itu, translasi alamat yang diterapkan dalam NAT, membuat para *cracker* di Internet tidak mungkin menyerang langsung sistem-sistem di dalam jaringan internal. *Intruder* harus menyerang dan memperoleh akses ke mesin NAT dulu sebelum menyiapkan serangan ke mesin-mesin di jaringan internal. Penting di ketahui bahwa, sementara dengan NAT jaringan internal terproteksi, namun untuk masalah *security*, tetap saja diperlukan paket *filtering* dan metoda pengamanan lainnya dalam mesin NAT.

3.3. Instalasi Natd

Sebuah perusahaan kecil memiliki sejumlah komputer dan sambungan ke Internet. Komputer-komputer itu saat ini telah membentuk suatu LAN. Sambungan Internet-nya diasumsikan berupa *dedicated T1 link*

Langkah-langkah yang harus dilakukan

1. Instalasi FreeBSD

Sediakan satu komputer untuk dijadikan *Gateway*. Penulis menyarankan penggunaan **FreeBSD RELEASE 2.2.6** (Natd hanya jalan di FreeBSD 2.2.1 ke atas), karena selain gratis juga *requirement hardware*-nya tidak terlalu boros. PC 486 dengan 16 MB *memory* dan HD 850 MB juga sudah cukup mewah.

Untuk mengetahui proses instalasi FreeBSD, silahkan baca kembali tulisan-tulisan di Infokomputer sebelumnya dan manual FreeBSD sendiri.

2. Instalasi Gateway

Pasang 2 *network interface* agar mesin ini menjadi *gateway*. *Network Card* (misal NE2000 atau 3COM) satu dihubungkan ke jaringan internal dan satu lagi untuk koneksi ke ISP. Misalnya dua-duanya NE2000 *Compatible*. maka *nick* untuk *card* yang menghadap ke dalam adalah ed0 dan untuk card yang menghadap keluar adalah ed1.

Pastikan juga option gateway = "YES" tertulis dengan benar dalam *file rc.conf*. Atau bisa juga dengan mengetik perintah: `sysctl -w net.inet.ip.forwarding=1`

3. Instalasi Firewall

Pasang IP *firewall* di mesin FreeBSD ini. Caranya adalah :

- a. Edit *kernel source* di `/usr/src/sys/i386/conf`

Tambahkan *option-option* berikut ini pada file *kernel*.

options IPFIREWALL

options IPFIREWALL_VERBOSE

options "IPFIREWALL_VERBOSE_LIMIT=100"

options IPDIVERT

- b. Compile *kernel* tersebut
- c. Aktifkan *firewall* di *rc.conf* dengan menambahkan
`firewall="YES"`
`firewall_type="OPEN"`

4. **Instalasi Natd**

Langkah-langkahnya adalah sbb:

- a. *Download source* nya di `ftp://ftp.suutari.iki.fi/pub/natd`
- b. *Unzip* dan *untar archive* tersebut dengan perintah
`gzip -dc natd_1.12.tar.gz | tar -xvf -`
- c. Lakukan *make* dan *make install* di direktori yang dihasilkan. Ketikkan perintah berikut:
`cd natd_1.12`
`make`
`make install`
- d. Edit *startup file* supaya Natd berjalan secara otomatis
 Buat file `natd.sh` di `/usr/local/etc/rc.d`. Isi file tersebut adalah
`#!/bin/sh`
`/sbin/ipfw -f flush`
`/sbin/ipfw add divert 13494 ip from any to any via ed0`
`/sbin/ipfw add pass all from 127.0.0.1 to 127.0.0.1`
`/sbin/ipfw add pass ip from any to any`
`/usr/local/sbin/natd -port 13494 -interface ed0`

Arti dari file ini adalah:

- ❖ Hapuskan semua rule *firewall*
- ❖ Tambahkan feature *divert* di *port* 13494 (Anda bisa mengganti ini dengan *port* yang Anda inginkan) untuk mendiversi paket dari dan ke *gateway* lewat *interface* `ed0`
- ❖ Bolehkan semua paket lewat di atas local host

- ❖ Bolehkan semua paket IP lewat semua *interface*
 - ❖ Jalankan Natd dengan menjadi *daemon* yang menunggu di *port* 13494 via *interface* ed0.
- e. Reboot mesin FreeBSD-nya supaya setting bisa diaktifkan.

3.4. Konfigurasi TCP/IP Client.

Jadikan nomor IP *card* ed0 di FreeBSD sebagai *gateway* dari tiap *workstation*, IP tiap-tiap *work station* harus berada dalam *network* yang sama dengan *card* ed0 yang ada di mesin *gateway*. Misal ed0 di-beri nomor IP 192.168.1.1 dan ed1 167.205.19.5, maka *workstation* diberi nomor IP 192.168.1.2 s/d 192.168.1.14 jika digunakan *mask* 16 atau 255.255.255.240. ed1 adalah *interface* yang memiliki IP *address* valid. Setelah semuanya langkah-langkah di atas dijalankan dengan baik maka, aplikasi Internet di *client* siap dijalankan via NAT.

Untuk kasus lain misalnya sambungan ke Internet-nya menggunakan modem, maka mekanismenya sama saja, tinggal diganti *interface* di *gateway* yang menghadap keluar dengan *interface* modem (tun0) dan jalankan program ppp untuk men-*dial* ISP-nya. Khusus untuk *dial-out*, ppp sebenarnya memiliki mekanisme sendiri untuk kasus ini yaitu dengan option -alias. Jadi jika kita menjalankan ppp dengan option -alias maka kita tidak perlu menjalankan Natd, karena option ini menyediakan fasilitas yang sama dengan Natd khusus untuk *dial-out*.

BAB IV PENUTUP

Dari pembahasan diatas pada bab-bab sebelumnya maka dapat disimpulkan bahwa Logika sederhana untuk penghematan IP *address* ialah dengan meng-*share* suatu nomor IP *address* valid ke beberapa *client* IP lainnya. Salah satu mekanisme itu disediakan oleh *Network Address Translation* (NAT)

Natd hanyalah salah satu cara untuk menghemat persediaan IP *address* yang semakin menipis. Dengan adanya fakta bahwa untuk bergabung ke Internet, *host* pencari informasi (*Client*) sebenarnya tidak perlu memiliki IP *address* legal, maka IP *address* legal tersebut bisa dicadangkan untuk *host-host* penyedia informasi (*Server*).

Model IP versi baru yaitu IP versi 6 (IPv6), bisa menampung lebih banyak lagi komputer-komputer di Internet. Namun demikian untuk kondisi-kondisi tertentu, Natd merupakan salah satu solusi ampuh selain menerapkan IPv6 diterapkan.

DAFTAR PUSTAKA

- Douba, Salim, 1995, *Networking UNIX, The Complete Reference for UNIX Networks*, SAMS Publishing.
- FreeBSD Handbook*, 1998, FreeBSD Inc.
- Tito@TI.ITB.ac.id, 2005.
- Unix Integration to WAN*, 2000, : *Applied Computer Internetworking*
CNRG ITB.